UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,359	09/25/2003	Charles Zdzislaw Loboz	614-L	7112

27201 7590 07/22/2008

UNISYS CORPORATION
Unisys Way
Mail Station E8-114
Blue Bell, PA 19424

EXAMINER	
TIMBLIN, ROBERT M	
ART UNIT	PAPER NUMBER
2167	

MAIL DATE	DELIVERY MODE
07/22/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/671,359	LOBOZ ET AL.	
	Examiner	Art Unit	
	ROBERT TIMBLIN	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 March 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *21-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *21-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This Office Action corresponds to application 10/671,359 filed 9/25/2003.

Response to Amendment

In the present Application, claims 25-27 are newly added and are pending. All previous claims have been cancelled at the request of the Applicant. Accordingly claims 21-27 are currently under examination.

Claim Objections

Examiner thanks the Applicant for consideration and response to the previous claim objections. Accordingly, in light of the minor corrections and remarks, the claim objections have been withdrawn.

However, claim 26 is newly objected to because there should be a comma (,) after "memory means" in and "an" preceding "electronic database" the first line as to clarify readability of the claim.

Claim Rejections - 35 USC § 112

In light of Applicant's remarks, the previous 112 rejections have been withdrawn.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 23-24 are rejected under 35 U.S.C. 101 because they may be construed to incorporate software per se (i.e. a program), thus being functional descriptive material per se and not statutory. As the means in these claims are unclear as to what they define, the can be interpreted as software per se (i.e. page 9, lines 19-23 describe that the present invention may be executed on any suitable computing system, with any suitable hardware and/*or software)*. MPEP 2106.01 states computer programs claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized.

In response to Applicant's remarks (page 8-9 of the reply), it is recited that claims 23-24 are now amended to be understood as physical modules and physical elements. The Examiner respectfully disagrees and asserts that the claims may still be interpreted as pure software embodidiments. That is, (as noted above in regards to page 9, lines 29-23) the "apparatus" may be construed as a software apparatus because the means that define it may be seen as software means. Also, the Examiner submits that "operating in a computer system" is insufficient to describe hardware because the claim is directed towards the apparatus comprising a means and

not elements of the computer system. Further, although Applicant points to hardware elements in the remarks (e.g. middle of page 9), these elements are not clearly defined in claims 23-24. In other words, the apparatus of claim 23 does not explicitly recite containing and being formed of any hardware elements. With the Applicant indicating that one implementation may be hardware (last paragraph of page 8) and this implementation is not recited in the claims, they are maintained to be software (i.e. the other possible implementation).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 21-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krychniak (U.S. Patent 6,192,357) in view of Prabhakaran et al. ("Prabhakaran" hereafter) (U.S. Patent 6,859,758) and further in view of Tenorio et al. ("Tenorio" hereafter) (U.S. Patent 6,708,161).

With respect to claim 21, Krychniak teaches A method operative in a computer system for improving the performance of a database by determining whether or not to alter the fields of the database having entities which hold a set of data values, said database including at least one set of linked entities, wherein the at least one set of linked entities contains a plurality of conceptual

entities, each of the conceptual entities including a plurality of data values which are distributed amongst the plurality of the conceptual entities, comprising the steps of:

(iiia) defining an additional entity table (figure 1, fact table) in the database (drawing reference 6, figure 3); and

(iiib) storing in the additional entity table (figure 1, fact table) an aggregation of said plurality of data values (i.e. fact table contains the keys of the dimensional tables) representing an aggregation of at least one of the plurality of conceptual entities (dimensions 1-3, figure 1 and figures 2a-2c), whereby the information defining the conceptual entity is obtained by performing a single read operation (co. 2 lines 11-28 discloses performing a read operation on the fact table; see also figure 4 wherein in certain situations, use of a join query is avoided) on the additional entity table (figure 1, fact table).

Krychniak fails to expressly teach step (i) of determining an average read/write ratio of the plurality of data values and (ii) comparing the average read/write ratio to a predetermined critical read/write ratio.

Prabhakaran, however, teaches step (i) of determining an average read/write ratio (col. 5 line 55-65; e.g. a approximate desired ratio) of the plurality of data values (col. 6 line 27-29) to estimate the performance of the database. Further, Prabhakaran teaches (ii) comparing the average read/write (step 330, col. 5 lien 66-67, and col. 6 line 31-47) to suggest comparing the average read/write ratio with other read to write ratios.

In the same field of endeavor, (i.e. query/access performance on differing database implementations), it would have been obvious to one of ordinary skill in the data processing art

at the time of the present invention to combine the teachings of the cited references because an average read/write ratio from Prabhakaran would have given Krychniak the benefit of an efficient threshold on which to determine a specific access method (Krychniak, figure 4 and Krychniak's claim 3). Further, the comparison as taught by Prabhakaran would have given Krychniak a way to compare access performance to decide which access method to perform (needed by Krychniak in figure 4).

Krychniak and Prabhakaran fail to expressly teach in step (ii) a predetermined critical read/write ratio.

Tenorio, however, teaches a critical read/write ratio (col. 17 line 55-59, and figure 6) for comparing implementations (i.e. indexed or not indexed) of a database.

In the same field of endeavor, (i.e. determining access performance), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Tenorio would have given Krychniak/Prabhakaran a ratio with which to compare performance of access in different implementations for the benefit of choosing an efficient database implementation. Furthermore, the comparison of the average read/write ratio (desired ratio of Prabhakaran) to Tenorio's critical read/write ratio would have given Krychniak a basis on which to define a fact table. This could have been done if (applicant's step (iii)) the average read/write ratio is greater than the critical read/write ratio (i.e. Prabhakaran tries to achieve the desired ratio and therefore the desired (average) ratio is suggested to be greater than the critical ratio) for the benefit of choosing an efficient database implementation.

With respect to claim 22, the combination of Krychniak and Prabhakaran fail to explicitly teach the steps (iia-iie) for predetermining a critical read/write ratio.

Tenorio, however, teaches wherein the step (ii) for predetermining the critical read/write ratio includes the further steps of:

(iia) providing data with regard to the time taken (figure 3, i.e. selecting a time period) to perform a read operation (figure 6, drawing reference 202) and a write operation (drawing reference 204) on a first implementation (e.g. an implementation that is indexed) of the said database (drawing reference 32);

(iib) providing data with regard to the time taken to perform a read operation (figure 6, drawing reference 202) and a write operation (drawing reference 204) on a second implementation (e.g. an implementation without an index) of the said database (drawing reference 32);

calculating a read time difference (drawing reference 216 and col. 17 line 45-59) between the time taken to perform a read operation on said first implementation (e.g. an implementation that is indexed) of said database and on said second implementation (e.g. an implementation without an index) of said database (drawing reference 32);

(iid) calculating a write time difference (drawing reference 216 and col. 17 line 45-59) between the time taken to perform a write operation on said first implementation (e.g. an implementation that is indexed) of said database and on said second implementation (e.g. an implementation without an index) of said database (drawing reference 32); and

(iic) calculating the ratio between the read time difference and the write time difference (figure 6 and drawing reference 218) to determine the critical read/write ratio for the database (drawing reference 32).

Accordingly, it would have been obvious to one of ordinary skill in the art at the time of the present invention to compare the total read and write times for each implementation of the database to provide a critical read/write ratio to compare to the average read/write ratio (taught by Prabhakaran). By doing so, a combination of the prior art would give Krychniak an efficient way of choosing an implementation (i.e. using a fact table or not) to optimize accessing efficiency.

With respect to claim 23, Krychniak teaches An apparatus operating in a computer system for modifying a database by determining whether or not to alter the fields of the database having entities which hold a set of data values, comprising:

(a) means for providing at least one set of linked entities (figure 1, dimensions and col. 1 line 26) in the said database (drawing reference 6), wherein the said at least one set of linked entities (dimensions) contains a plurality of conceptual entities (dimensions 1-3, figure 1 and figures 2a-2c), each of the conceptual entities (dimensions) including a plurality of data values (figure 1, attributes and keys) which are distributed amongst (abstract) the plurality of conceptual entities (dimensional);

(d1) means for defining an additional entity table (figure 1, fact table) in addition to the at least one set of linked entities (dimensions);

(d2) storing means arranged to store, in said additional entity table (figure 1, fact table), the aggregation of said plurality of data values (i.e. fact table contains the keys of the dimensional table) representing an aggregation of at least one said plurality of conceptual entities (dimensions 1-3, figure 1 and figures 2a-2c); and

(d3) reading means enabled to read said aggregation of said plurality of data values by performing a single read operation (co. 2 lines 11-28 discloses performing a read operation on the fact table; see also figure 4 wherein in certain situations, use of a join query is avoided) on said additional entity table (fact table) to return the information determining at least one conceptual entity (col. 2 line 9-10).

Krychniak fails to expressly teach determining an average read/write ratio of the plurality of data values and comparing the average read/write ratio to a predetermined critical read/write ratio.

Prabhakaran, however, teaches determining an average read/write ratio (col. 5 line 55-65; e.g. a approximate desired ratio) of the plurality of data values (col. 6 line 27-29) to estimate the performance of the database. Further, Prabhakaran teaches comparing the average read/write (step 330, col. 5 lien 66-67, and col. 6 line 31-47) to suggest comparing the average read/write ratio with other read to write ratios.

In the same field of endeavor, (i.e. query/access performance on differing database implementations), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because an average read/write ratio from Prabhakaran would have given Krychniak the benefit of an

efficient threshold on which to determine a specific access method (Krychniak, figure 4 and Krychniak's claim 3). Further, the comparison as taught by Prabhakaran would have given Krychniak a way to compare access performance to decide which access method to perform (needed by Krychniak in figure 4).

Krychniak and Prabhakaran fail to expressly teach a predetermined critical read/write ratio.

Tenorio, however, teaches a critical read/write ratio (col. 17 line 55-59, and figure 6) for comparing implementations (i.e. indexed or not indexed) of a database.

In the same field of endeavor, (i.e. determining access performance), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Tenorio would have given Krychniak/Prabhakaran a ratio with which to compare performance of access in different implementations for the benefit of choosing an efficient database implementation. Furthermore, the comparison of the average read/write ratio (desired ratio of Prabhakaran) to Tenorio's critical read/write ratio would have given Krychniak a basis on which to define a fact table. This could have been done if (applicant's step (d)) the average read/write ratio is greater than the critical read/write ratio (i.e. Prabhakaran tries to achieve the desired ratio and therefore the desired (average) ratio is suggested to be greater than the critical ratio) for the benefit of choosing an efficient database implementation.

With respect to claim 24, the combination of Krychniak and Prabhakaran fail to explicitly teach the steps (ca-ce) for predetermining a critical read/write ratio:

Tenorio, however, teaches steps (ca-ce) for predetermining a critical read/write ratio as:

(ca) means for providing data with regard to the time taken (figure 6, i.e. selecting a time period) to perform a read operation (figure 6, drawing reference 202) and a write operation (drawing reference 204) on the data values which are distributed amongst the plurality of entities;

(cb) means for providing data with regard to the time taken (figure 6, i.e. selecting a time period) to perform a read operation (figure 6, drawing reference 202) and a write operation (drawing reference 204) on said additional entity table;

(cc) means for calculating a read time difference (drawing reference 216 and col. 17 line 45-59) between the time taken to perform a read operation (figure 6, drawing reference 202) on the data values which are distributed amongst the plurality of entities (Krychniak, dimension) and on said additional entity table (Krychniak, fact table);

(cd) means for calculating a write time difference (drawing reference 216 and col. 17 line 45-59) between the time taken to perform a write operation on the data values which are distributed amongst the plurality of entities (Krychniak, dimension) and on said additional entity table (Krychniak, fact table); and

(ce) means for calculating the ratio between said read time difference (figure 6 and drawing reference 218) and said write time difference to determine the critical read/write ratio for the database (drawing reference 32).

Accordingly, it would have been obvious to one of ordinary skill in the art at the time of the present invention to compare the total read and write times for each implementation of the database to provide a critical read/write ratio to compare to the average read/write ratio (taught by Prabhakaran). By doing so, a combination of the prior art would give Krychniak an efficient way of choosing an implementation (i.e. using a fact table or not) to optimize accessing efficiency.

With respect to claim 25, this claim incorporates the limitations found in claim 21 and therefore is rejected for the same reasons.

With respect to claim 26, Krychniak teaches A machine holding a computer, memory means and electronic database which utilizes a computer program for improving the performance of said database including at least one set or linked entities, wherein the at least one set of linked entities contains a plurality of conceptual entities, each of the conceptual entities including a plurality of data values which are distributed amongst the plurality of the conceptual entities, said computer program including at least one instruction which, when executed by a computer system, is arranged to carry .out the following steps:

(iiia) defining an additional entity table (figure 1, fact table) in the database (drawing reference 6, figure 3); and

(iiib) storing in the additional entity table (figure 1, fact table) an aggregation of said plurality of data values (i.e. fact table contains the keys of the dimensional table) representing an

aggregation of at least one of the plurality of conceptual entities (dimensions 1-3, figure 1 and figures 2a-2c), whereby the information defining the conceptual entity is obtained by performing a single read operation (co. 2 lines 11-28 discloses performing a read operation on the fact table; see also figure 4 wherein in certain situations, use of a join query is avoided) on the additional entity table (figure 1, fact table).

Krychniak fails to expressly teach step (i) of determining an average read/write ratio of the plurality of data values and (ii) comparing the average read/write ratio to a predetermined critical read/write ratio.

Prabhakaran, however, teaches step (i) of determining an average read/write ratio (col. 5 line 55-65; e.g. a approximate desired ratio) of the plurality of data values (col. 6 line 27-29) to estimate the performance of the database. Further, Prabhakaran teaches (ii) comparing the average read/write (step 330, col. 5 lien 66-67, and col. 6 line 31-47) to suggest comparing the average read/write ratio with other read to write ratios.

In the same field of endeavor, (i.e. query/access performance on differing database implementations), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because an average read/write ratio from Prabhakaran would have given Krychniak the benefit of an efficient threshold on which to determine a specific access method (Krychniak, figure 4 and Krychniak's claim 3). Further, the comparison as taught by Prabhakaran would have given Krychniak a way to compare access performance to decide which access method to perform (needed by Krychniak in figure 4).

Krychniak and Prabhakaran fail to expressly teach in step (ii) a predetermined critical read/write ratio.

Tenorio, however, teaches a critical read/write ratio (col. 17 line 55-59, and figure 6) for comparing implementations (i.e. indexed or not indexed) of a database.

In the same field of endeavor, (i.e. determining access performance), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Tenorio would have given Krychniak/Prabhakaran a ratio with which to compare performance of access in different implementations for the benefit of choosing an efficient database implementation. Furthermore, the comparison of the average read/write ratio (desired ratio of Prabhakaran) to Tenorio's critical read/write ratio would have given Krychniak a basis on which to define a fact table. This could have been done if (applicant's step (iii)) the average read/write ratio is greater than the critical read/write ratio (i.e. Prabhakaran tries to achieve the desired ratio and therefore the desired (average) ratio is suggested to be greater than the critical ratio) for the benefit of choosing an efficient database implementation.

With respect to claim 27, the combination of Krychniak and Prabhakaran fail to explicitly teach the machine of claim 26 which utilizes the said computer program for improving the performance of the database, wherein in step (ii), the predetermined critical read/write ratio is calculated by carrying out the follow steps:

Tenorio, however, teaches wherein the step (ii) for predetermining the critical read/write ratio includes the further steps of:

(iia) providing data with regard to the time taken (figure 3, i.e. selecting a time period) to perform a read operation (figure 6, drawing reference 202) and a write operation (drawing reference 204) on a first implementation (e.g. an implementation that is indexed) of the said database (drawing reference 32);

(iib) providing data with regard to the time taken to perform a read operation (figure 6, drawing reference 202) and a write operation (drawing reference 204) on a second implementation (e.g. an implementation without an index) of the said database (drawing reference 32);

calculating a read time difference (drawing reference 216 and col. 17 line 45-59) between the time taken to perform a read operation on said first implementation (e.g. an implementation that is indexed) of said database and on said second implementation (e.g. an implementation without an index) of said database (drawing reference 32);

(iid) calculating a write time difference (drawing reference 216 and col. 17 line 45-59) between the time taken to perform a write operation on said first implementation (e.g. an implementation that is indexed) of said database and on said second implementation (e.g. an implementation without an index) of said database (drawing reference 32); and

(iie) calculating the ratio between the read time difference and the write time difference (figure 6 and drawing reference 218) to determine the critical read/write ratio for the database (drawing reference 32).

Accordingly, it would have been obvious to one of ordinary skill in the art at the time of the present invention to compare the total read and write times for each implementation of the database to provide a critical read/write ratio to compare to the average read/write ratio (taught by Prabhakaran). By doing so, a combination of the prior art would give Krychniak an efficient way of choosing an implementation (i.e. using a fact table or not) to optimize accessing efficiency.

Response to Arguments

Applicant's arguments filed in the reply dated 3/12/2008 have been fully considered but they are not persuasive.

On page 8-9 of the reply, Applicant argues the previous 35 U.S.C. 112 1st paragraph rejection. In light of the remarks, this rejection is withdrawn.

On page 10, top portion of the reply, Applicant responds to the previous 35 U.S.C. 101 rejection in regards to the claims being software. The Examiner has noted the response and replied accordingly in the respective 101 heading of this document (see above).

In the bottom portion of page 10 through page 14, Applicant argues the present claim rejections under 35 U.S.C. 103(a). The arguments have been noted, however, are found unpersuasive given the following:

With regards to a first point, Applicant argues that Krychniak does not teach defining an additional entity table which stores an aggregation of data values. The Examiner disagrees and maintains that Krychniak teaches this aspect. That is, Krychniak teaches a fact table (see figure 1) that have columns that represent a particular dimension (col. 1 lines 13-15). In other words, the keys which are stored in dimension tables 1-3 are respectively stored in the fact table, which is clearly shown in figure 1 and described in col. 1 line 39-41. That is, the actual keys from the dimensions are stored in the fact table rather than arbitrary identifiers, which Applicant misinterprets (page 11 of the remarks indented portion) Krychniak as teaching. Further, because the keys of the dimensions are stored in the fact table, the fact table therein represents an aggregation (i.e. a collection of the keys) of data values (e.g. keys) and thus at least describes and suggests the claimed an additional entity table (storing an aggregation of data values).

Applicant then argues (page 11, 2nd paragraph) that they are puzzled how the passages teach interrogating the database to determine read/write ratios. In response to Applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., interrogating) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In other words, the claim does not recite interrogating a database, however it does plainly recite "determining" an average read/write ratio of data values in a database. The Examiner submits, as noted in the above rejection, that this ratio is found in Prabhakaran. Further, even *if*

Prabhakaran does *set* the ratio, under a reasonably broad interpretation, one would understand that the ratio that is set for a database is essentially the same as a ratio that is "determined" for a database. Furthermore, one would also understand that to obtain an average read/write ratio for a database that the database itself would have to be tested and performed upon to arrive at an average. In either case, the Examiner submits that an average read/write ratio is determined for a database as described by Prabhakaran.

Applicant also argues that there is not explicit teaching of comparing ratios (e.g. page 12, top indented paragraph) in Prabhakaran. The Examiner respectfully disagrees because although Prabhakaran does not explicitly recite *comparing*, that this element is not without description in the reference. That is, Prabhakaran discloses trying to achieve a given ratio. Further, Prabhakaran discloses trying to find a *correspondence* between read/write operations to a desired ratio. Both of these elements are found in Prabhakaran, drawing references 320 and 330. In other words when a set of read/write operations are attempting to achieve another ratio (i.e. "achieve" suggests a comparison) that this element in Applicant's claims is met.

On a further note and another example of how the applied references teach the "comparing" language, Tenorio also teaches a comparison of ratios (e.g. col. 17 line 45-59 and figure 6) in that the read/write operations for one implementation are compared to the read/write operations for another implementation.

As a second point, Applicant argues (page 12 of the reply) that there is no suggestion and motivation to combine the references. In response to applicant's argument that there is no

suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Examiner finds suggestion and motivation as noted in the rejection above. Further, the Examiner submits that Krychniak does suggest a need to modify their database structure. For example, Krychniak shows this by disclosing if they should consider indexing using a fact table (e.g. Krychniak, col. 2 line 29-36; "databases *might* be optimized for example by indexing the fact table…*if* such an index based optimizing scheme is used…). Furthermore, with the combination of the comparison of Prabhakaran's ratio with Tenorio's, Krychniak would benefit by having a way to determine if they should index to optimize data retrieval. Furthermore, since all three applied references are analogous (e.g. dealing with information retrieval and optimizing) that one of ordinary skill would be motivated to combine the references for the benefit of faster and more efficient data retrieval.

Related Prior Art

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

6,088,767 to Dan et al. The subject matter disclosed therein pertains to the pending claims (i.e. read/write ratios).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert M. Timblin whose telephone number is 571-272-5627. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/ROBERT TIMBLIN/
Examiner, Art Unit 2167

/John R. Cottingham/

Supervisory Patent Examiner, Art Unit 2167